Ahmat Barkai Moussa

Task 6 : Oluşturulan tablo için facade yazılması. Yeni tablo için yazılan facade ile birlikte populator yazılması ve converter ile Modelin Dataya convert edilmesi işlemini yapın. Homepage'de bir form oluşturun ve HomepageController'da bu facade'a istek atarak veriyi getirin, veriyi güncellemek için bir servis yazın ve formdan gelen verileri database'e kaydedin.

Avansasfacade-beans.xml'de ekliyip ant build yapıyoruz.

Data sınıfımız oluştu.

Sonradan doa interface ve class oluşturuyoruz. Query ve gereken metotlarımızı yaziyoruz.

Dao ve service injection için gerken beans'leri avansascore-spring.xml'de yazıyoruz.

Facade, populator ve reverse populator için beans'leri yazıyoruz.

Sonra service interface ve servisi Imp yazılıyor.

```
| import com.avansas.facades.product.TrainingProductFacade;
| import javax.annotation.Resource;
| import javax.annotation.Resource |
|
```

```
import com.avansas.facades.product.TrainingProductFacade;
import de.hybris.platform.servicelayer.dto.converter.Converter;

import javax.annotation.Resource;

import javax.dti.List;

lusage

public class TrainingProductFacadeImp implements TrainingProductFacade {

duages

@Resource(name="trainingProductService")

private TrainingProductService trainingProductService;

3 usages

@Resource(name="trainingProductConverter")

private Converter<TrainingProductReverssConverter")

private Converter<TrainingProductReverssConverter")

private Converter<TrainingProductBata_TrainingProductModel> trainingProductReverssConverter;

@Override

public List<TrainingProductConverter.convertAll(trainingProductService.getAllProductS());

lusage

@Override

public TrainingProductConverter.convertAll(trainingProductData trainingProductData) {

TrainingProductConverter.underTrainingProduct(TrainingProductBaverssConverter.convert(trainingProductData);

trainingProductConverter().convert(trainingProductModel);

return getTrainingProductConverter().convert(trainingProductModel);

return getTrainingProductConverter().convert(trainingProductModel);

lusage

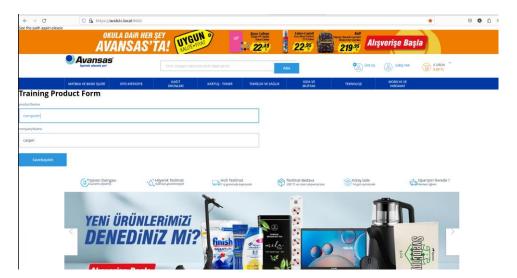
public Converter<TrainingProductConverter()

public Converter<TrainingProductConverter()
```

Facade'da service'teki metotları çağırıyoruz, populator yada reverse populator işlemleri yapıyoruz.

HomePageController'de verileri alabilmemiz için Get metodu ve post metodu yazıp pageModel döndürüyoruz.

Ve Front end 'de formumuzu ekleyip controller'de elde ettiğimiz (productName ve companyName) verileri form alanlarına ekliyoruz .



Ön yüzü formumuz verileriyle başarılıyla geldi ve ayna anda güncelleyebilmektedir.

